Your search matched **94** of **955082** documents.

A maximum of **94** results are displayed, **15** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text box.

Then click **Search Again**.

((object)and (event)) and(message)

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **Enhancing object communication mechanisms**
Ersavas, T.;
Technology of Object-Oriented Languages, 1998. TOOLS 27. Proceedings , 22-25 Sept. 1998
Page(s): 232 -244

[Abstract] [PDF Full-Text (276 KB)] **IEEE CNF**

2 **An application of a protective relaying scheme over an ethernet LAN/WAN**
Brunello, G.; Smith, R.; Campbell, C.B.;
Transmission and Distribution Conference and Exposition, 2001 IEEE/PES , Volume: 1 , 28 Oct.-2 Nov. 2001
Page(s): 522 -526 vol.1

[Abstract] [PDF Full-Text (239 KB)] **IEEE CNF**

3 **Exploiting component/event-level parallelism in concurrent fault and design error simulation**
Shaikh, S.A.; Szygenda, S.A.;
Simulation Symposium, 1997. Proceedings. 30th Annual , 7-9 April 1997
Page(s): 64 -74

[Abstract] [PDF Full-Text (996 KB)] **IEEE CNF**

4 **The flying object for an open distributed environment**

Oda, K.; Tazuneki, S.; Yoshida, T.;
Information Networking, 2001. Proceedings. 15th International Conference on , 31 Jan.-2 Feb. 2001
Page(s): 87 -92

[Abstract] [PDF Full-Text (536 KB)] **IEEE CNF**

---

5 **pp-mess-sim: a flexible and extensible simulator for evaluating multicomputer networks**
Rexford, J.; Wu-Chang Feng; Dolter, J.; Shin, K.G.;
Parallel and Distributed Systems, IEEE Transactions on , Volume: 8 Issue: 1 , Jan. 1997
Page(s): 25 -40

[Abstract] [PDF Full-Text (724 KB)] **IEEE JNL**

---

6 **Event ordering in a shared memory distributed system**
Gunaseelan, L.; LeBlanc, R.J., Jr.;
Distributed Computing Systems, 1993., Proceedings the 13th International Conference on , 25-28 May 1993
Page(s): 256 -263

[Abstract] [PDF Full-Text (652 KB)] **IEEE CNF**

---

7 **Enhancing an event-based OO framework for distributed programming**
Silverajan, B.; Harju, J.;
Technology of Object-Oriented Languages and Systems, 1999. TOOLS 30. Proceedings , 1-5 Aug. 1999
Page(s): 162 -171

[Abstract] [PDF Full-Text (116 KB)] **IEEE CNF**

---

8 **An asynchronous message exchange system on CORBA**
Tswen-Yuh Hsiao; Win-Tsung Lo; Shyan-Ming Yuan;
Technology of Object-Oriented Languages and Systems, 2000. TOOLS-Pacific 2000. Proceedings. 37th International Conference on , 20-23 Nov. 2000
Page(s): 14 -23

[Abstract] [PDF Full-Text (648 KB)] **IEEE CNF**

---

9 **Reliable messaging using the CORBA Notification Service**
Ramani, S.; Dasarathy, B.; Trivedi, K.S.;

Distributed Objects and Applications, 2001. DOA '01. Proceedings. 3rd International Symposium on , 17-20 Sept. 2001
Page(s): 229 -238

[Abstract] [PDF Full-Text (832 KB)] **IEEE CNF**

10 **Logical clock requirements for reverse engineering scenarios from a distributed system**
Hrischuk, C.E.; Woodside, C.M.;
Software Engineering, IEEE Transactions on , Volume: 28 Issue: 4 , April 2002
Page(s): 321 -339

[Abstract] [PDF Full-Text (482 KB)] **IEEE JNL**

11 **Implementation of the SEMI `Generic Equipment Model' using object-based cell technology**
DeBolt, J.R.; Wickizer, C.R.;
Semiconductor Manufacturing Science Symposium, 1991. ISMSS 1991., IEEE/SEMI International , 20-22 May 1991
Page(s): 102 -105

[Abstract] [PDF Full-Text (308 KB)] **IEEE CNF**

12 **Abstract simulator for the parallel DEVS formalism**
Chow, A.C.; Zeigler, B.P.; Doo Hwan Kim;
AI, Simulation, and Planning in High Autonomy Systems, 1994. 'Distributed Interactive Simulation Environments'., Proceedings of the Fifth Annual Conference on , 7-9 Dec. 1994
Page(s): 157 -163

[Abstract] [PDF Full-Text (472 KB)] **IEEE CNF**

13 **Conservative synchronization in object-oriented parallel battlefield discrete event simulations**
Hiller, J.B.; Hartrum, T.C.;
Parallel and Distributed Simulation, 1997. Proceedings. 11th Workshop on , 10-13 June 1997
Page(s): 12 -19

[Abstract] [PDF Full-Text (888 KB)] **IEEE CNF**

14 **Execution-driven simulation of error recovery techniques for multicomputers**

Frazier, T.M.; Tamir, Y.;
Simulation Symposium, 1997. Proceedings. 30th Annual , 7-9 April 1997
Page(s): 4 -13

[Abstract] [PDF Full-Text (996 KB)] **IEEE CNF**

---

15 **A framework for event-driven demonstration based on the Java toolkit**
Miura, M.; Tanaka, J.;
Computer Human Interaction, 1998. Proceedings. 3rd Asia Pacific , 15-17 July 1998
Page(s): 331 -336

[Abstract] [PDF Full-Text (88 KB)] **IEEE CNF**

---

1 2 3 4 5 6 7 [Next]

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Search Results for: **[GUI<AND>((message<AND>((event<AND>((object))))))]**
Found **727** of **114,152 searched.**

Warning: Maximum result set of 200 exceeded. Consider refining.

Search within Results

> Advanced Search
> Search Help/Tips

Sort by: Title Publication Publication Date Score

Results 1 - 20 of 200 short listing

Prev Page 1 2 3 4 5 6 7 8 9 10 Next Page

1 A proposal for calling conventions for APL defined functions 97%
Vincent Lin
ACM SIGAPL APL Quote Quad , Proceedings of the conference on Designing the future June 1996
Volume 26 Issue 4
This paper will touch on the lack of mechanisms to pass arguments back from defined functions other than return values, if return values are already used as error codes. An extreme case is the use of call-back functions, or event procedures, to the events of GUI objects. In Dyalog APL/W, functionality of arguments and return values of call-back functions are already well-defined. The way you can pass information back from a call-back function, or event procedure, is by storing it as a global var ...

2 An overview of HI-MASS (Hierarchical Modeling and Simulation System) 97%
Douglas G. Fritz , Robert G. Sargent , Thorsten Daum
Proceedings of the 27th conference on Winter simulation December 1995

3 A new approach to software tool interoperability 96%
Yimin Bao , Ellis Horowitz
Proceedings of the 1996 ACM symposium on Applied Computing February 1996

4 Multiple instances and symbolic variables in executable sequence charts 95%
Rami Marelly , David Harel , Hillel Kugler
ACM SIGPLAN Notices , Proceedings of the 17th ACM conference on Object-oriented

programming, systems, languages, and applications November 2002
Volume 37 Issue 11
We extend live sequence charts (LSCs), a highly expressive variant of sequence diagrams, and provide the extension with an executable semantics. The extension involves support for instances that can bind to multiple objects and symbolic variables that can bind to arbitrary values. The result is a powerful executable language for expressing behavioral requirements on the level of inter-object interaction. The extension is implemented in full in our *play-engine* tool, with which one can exec ...

5 FranTk - a declarative GUI language for Haskell 95%

Meurig Sage
ACM SIGPLAN Notices , Proceedings of the fifth ACM SIGPLAN international conference on Functional programming September 2000
Volume 35 Issue 9
FranTk is a new high level library for programming Graphical User Interfaces (GUIs) in Haskell. It is based on Fran (Functional Reactive Animation), and uses the notions of *Behaviors* and *Events* to structure code. Behaviors are time-varying, reactive values. They can be used to represent the state of an application. Events are streams of values that occur at discrete points in time. They can be used, for instance, to represent user input. FranTk allows user interfaces to be structur ...

6 Window real objects: a distributed shared memory for distributed implementation of GUI applications 94%

Noboru Koshizuka , Ken Sakamura
Proceedings of the 6th annual ACM symposium on User interface software and technology December 1993

7 Fast detection of communication patterns in distributed executions 94%

Thomas Kunz , Michiel F. H. Seuren
Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research November 1997
Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

8 Structuring graphical paradigms in TkGofer 94%

Koen Claessen , Ton Vullinghs , Erik Meijer
ACM SIGPLAN Notices , Proceedings of the second ACM SIGPLAN international conference on Functional programming August 1997
Volume 32 Issue 8
In this paper we describe the implementation of several graphical programming paradigms (Model View Controller, Fudgets, and Functional Animations) using the GUI library TkGofer. This library relies on a combination of monads and multiple-parameter type classes to provide an abstract, type safe interface to Tcl/Tk. We show how choosing the right abstractions makes the given implementations surprisingly concise and easy to understand.

9 Implementation of a diagnostic and troubleshooting multi-agent system for cellular networks 90%
Mahamat Guiagoussou , Said Soulhi
International Journal of Network Management August 1999
Volume 9 Issue 4
This article presents the implementation of a maintenance application for cellular switching system using the multi-agent paradigm. The main philosophy behind the design of the multi-agent system is based on the TMN framework, where each agent can mapped with one or several TMN functional blocks. Copyright © 1999 John Wiley & Sons, Ltd.

10 Programming languages as operating systems (or revenge of the son of the lisp machine) 89%
Matthew Flatt , Robert Bruce Findler , Shriram Krishnamurthi , Matthias Felleisen
ACM SIGPLAN Notices , Proceedings of the fourth ACM SIGPLAN international conference on Functional programming September 1999
Volume 34 Issue 9
The MrEd virtual machine serves both as the implementation platform for the DrScheme programming environment, and as the underlying Scheme engine for executing expressions and programs entered into DrScheme's read-eval-print loop. We describe the key elements of the MrEd virtual machine for building a programming environment, and we step through the implementation of a miniature version of DrScheme in MrEd. More generally, we show how MrEd defines a high-level operating system for graphical prog ...

11 Combining Ada 95, Java byte code, and the distributed systems annex 89%
Brad Balfour
Proceedings of the conference on TRI-Ada '97 November 1997

12 CircusTalk: an orchestration service for distributed multimedia 89%
Yechezkal Shimon Gutfreund , Jose Diaz-Gonzalez , Russell Sasnett , Vincent Phuah
Proceedings of the first ACM international conference on Multimedia September 1993

13 Technical papers: design recovery: Browsing and searching source code of applications written using a GUI framework 89%
Amir Michail
Proceedings of the 24th international conference on Software engineering May 2002
Nowadays, applications are typically written using an object-oriented GUI framework. In this paper we explore the possibility of using the GUI of such applications to guide browsing and search of their source code. Such a tool would be helpful for software maintenance and reuse, particularly when the application source is unfamiliar. Intuitively, we would expect the task of browsing and searching source code of an application written using a GUI framework to be easier than one that doesn't becau ...

14 Composable ad-hoc mobile services for universal interaction 88%
Todd D. Hodes , Randy H. Katz , Edouard Servan-Schreiber , Lawrence Rowe
Proceedings of the third annual ACM/IEEE international conference on Mobile computing and networking September 1997

15 Chiron-1: a software architecture for user interface development, maintenance, and run-time support 88%

Richard N. Taylor , Kari A. Nies , Gregory Alan Bolcer , Craig A. MacFarlane , Kenneth M. Anderson , Gregory F. Johnson

ACM Transactions on Computer-Human Interaction (TOCHI) June 1995
Volume 2 Issue 2

The Chiron-1 user interface system demonstrates key techniques that enable a strict separation of an application from its user interface. These techniques include separating the control-flow aspects of the application and user interface: they are concurrent and may contain many threads. Chiron also separates windowing and look-and-feel issues from dialogue and abstract presentation decisions via mechanisms employing a client-server architecture. To separate application code from user interf ...

16 FUDGETS: a graphical user interface in a lazy functional language 88%

Magnus Carlsson , Thomas Hallgren

Proceedings of the conference on Functional programming languages and computer architecture July 1993

17 Interaction techniques for ambiguity resolution in recognition-based interfaces 88%

Jennifer Mankoff , Scott E. Hudson , Gregory D. Abowd

Proceedings of the 13th annual ACM symposium on User interface software and technology November 2000

18 A scalable formal method for design and automatic checking of user interfaces 88%

Jean Berstel , Stefano Crespi Reghizzi , Gilles Roussel , Pierluigi San Pietro

Proceedings of the 23rd international conference on Software engineering July 2001

The paper addresses the formal specification, design and implementation of the behavioral component of graphical user interfaces. Dialogs are specified by means of modular, communicating grammars called VEG (Visual Event Grammars), which extend traditional BNF grammars to make the modeling of dialogs more convenient.

A VEG specification is independent of the actual layout of the GUI, but it can be easily integrated with various layout design toolkits. The specification may b ...

19 DeepView: a channel for distributed microscopy and informatics 88%

B. Parvin , J. Taylor , G. Cong , M. A. OKeefe , M. H. Barcellos-Hoff

Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM) January 1999

20 jRapture: A Capture/Replay tool for observation-based testing 88%

John Steven , Pravir Chandra , Bob Fleck , Andy Podgurski

ACM SIGSOFT Software Engineering Notes , Proceedings of the International Symposium on Software Testing and Analysis August 2000
Volume 25 Issue 5

We describe the design of jRapture: a tool for capturing and replaying Java program executions in the field. jRapture works with Java binaries (byte code) and any compliant implementation of

the Java virtual machine. It employs a lightweight, transparent capture process that permits unobtrusive capture of a Java programs executions. jRapture captures interactions between a Java program and the system, including GUI, file, and console inputs, among other types, and on replay it presents eac ...

Results 1 - 20 of 200 short listing

Prev Page 1 2 3 4 5 6 7 8 9 10 Next Page

CiteSeer Find: object and message and event and G Documents Citations

Searching for **object and message and event and gui**.
Restrict to: Header Title Order by: Citations Hubs Usage Date Try: Amazon B&N Google (RI) Google (Web) CSB DBLP
31 documents found. **Order: citations weighted by year.**

Robust State Sharing for Wide Area Distributed Applications - Topol, Ahamad, Stasko (1998) (Correct) (8 citations)
Mocha provides support for robust shared **objects** on heterogeneous platforms, and utilizes
www.gvu.gatech.edu/gvu/people/student/./Brad.Topol/replica.ps.Z

One or more of the query terms is very common - only partial results have been returned. Try Google (RI).

Functional Languages and Graphical User Interfaces - a review .. - Noble, Runciman (1994) (Correct) (13 citations)
: 40 3.2.9 Window **Objects** -Widgets :
uebb.cs.tu-berlin.de/papers/external/functional-state-guis/functional-IO-GUI.ps.gz

PARMON: A Comprehensive Cluster Monitoring System - Rajkumar, Mohan, Gopal (1998) (Correct) (5 citations)
and implemented using the state-of-the-art **object**-oriented, client-server, and Java computing
.Hardware Level .Operating System Level .**Message** Passing Interfaces Level .Language/Compiler
of the same resource PARMON allows to define **events** and its automatic triggering whenever **event**
www.dgs.monash.edu.au/~rajkumar/papers/parmon.ps.gz

Black-Box Reuse within Frameworks based on Visual.. - Wagner, Sluijmers.. (1996) (Correct) (6 citations)
Application frameworks allow structured reuse of **object**-oriented design and source code, provided that
framework [Wein92, Gamma92]**Event** handling and **message** passing between the framework's **objects** fall
the ETapplication framework [Wein92, Gamma92]**Event** handling and **message** passing between the
www.ifi.unizh.ch/staff/bwagner/VisProgComponent.ps.gz

Distributed Simulation for a Communication Protocol .. - Chun, Moser.. (Correct)
layers of the hierarchy are linked into a single **object** module executed by a single processor. Beneath
task due to the many possible executions and **message** orderings. In this paper we describe a
development environment based on a discrete-**event** simulator. We have used this development
alpha.ece.ucsb.edu/~wesc/jhpc.ps

Performance of OmniBroker, an Implementation of CORBA 2.0 - Oolan Zimmer (Correct)
is a freely available CORBA 2.0-compliant **object** request broker. We examine the performance of
over sockets benchmarks using similar size **messages**. Using a profiler, we found that OmniBroker
newly-allocated region for each call. The reactor's **event** handler will then call a dispatch method in
red.cs.uiuc.edu/papers/ob-perf.ps

Object-Oriented Network Programming - An Overview of CORBA - Schmidt (Correct)
Object-Oriented Network Programming An Overview of
creation, activation and **object** management -**Message** exchange between **objects** ffl **Event**ually, CORBA
ftp.kiae.su/pub/.1/unix/lang/c++/ACE/ACE-documentation/corba4.ps.gz

Applying Distributed Simulation to a Communication Protocol . . . - Chun, al. (Correct)
four hierarchical layers are linked into a single **object** module executed by a single processor. Underneath
task due to the many possible executions and **message** orderings. We describe a communication protocol
development environment based on a discrete-**event** simulator. We have used this development
alpha.ece.ucsb.edu/~wesc/paper.ps

PETSc 2.0 Users Manual - Revision Satish (Correct)
:117 12.2 Viewers: Looking at PETSc **Objects** :
info.mcs.anl.gov/pub/tech_reports/reports/ANL9511.ps.Z

A Report on the context of CORBA - De Jager (Correct)
because it consists mostly of data structures and **objects** on the server side, and client applications
www.up.ac.za/academic/skoolit/hsn/docs/report/NDJ96a.ps.gz

An Experiment with Electronic Logs - Waller (Correct)
can be accomplished by sending an **event** to an **object** from another script. Widgets can also react to
from another script. Widgets can also react to **messages** (**events**) which are sent up the **object** hierarchy.
a widget so that the widget can react to various **events** such as mouse or keyboard input. As mentioned
adwww.fnal.gov/www/icalepcs/abstracts/Postscript/wpo62.ps

Visualization of Interaction Patterns in Program Executions - Jerding (1996) (Correct)
most standard programming languages (especially **object**-oriented) support the transition of static design
made up of repeated sequences and subsequences of **messages**. While we were initially focusing on the
gulf of abstraction between low-level execution **events** such as function calls, and high-level design
www.gvu.gatech.edu/people/student/Dean.Jerding/./research.ps.Z

IMcast: An Object-Oriented Tool for Image Multicasting - Eric Kass (Correct)
IMcast: An **Object**-Oriented Tool for Image Multicasting Eric R.
paradigm and is designed to take advantage of the **message**-based Microsoft Windows operating system. Images
asynchronous task controller uses multiple **event** queues to coordinate interaction of different
ftp.cps.msu.edu/pub/crg/PAPERS/icmcs97.ps.gz

Using Design Patterns to Evolve System Software from UNIX.. - Schmidt, Stephenson (1995) (Correct)
types of **events** such as timers, synchronization **objects**, signals, or I/O operations. We recently ported
alternative mechanisms (such as shared memory vs. **message** passing) on different OS platforms. There are
siesta.cs.wustl.edu/~schmidt/DP-experience-95.ps.gz

A Smalltalk Memory Profiler and its Performance Enhancement - Jingyu Sun (Correct)
Performance has always been a concern with **object**-oriented software. Much attention has been
profiler, compared to a profiler which records all **message** sends. It meant that the profiler would have
the user through a graphical display. Allocation **events** were processed through a pipeline that filtered
www4.ncsu.edu/~efg/oopsla97.ps

Runtime Visualization of Computer Architecture Simulations - Kok, Pimentel, Hertzberger (Correct)
paper is presented. 2 Pearl Pearl[7, 6] is an **object**-oriented discrete-**event** simulation language
to C. **Objects** interact with each other by sending **messages**. A **message** consists of three parts: the
carol.wins.uva.nl/~andy/mermaid/docs/paid2.ps.gz

Implementation of the Dynamic Behavior of Object Oriented System - Ali, Tanaka (Correct)
Implementation of the Dynamic Behavior of **Object** Oriented System Jauhar Ali and Jiro Tanaka
www.softlab.is.tsukuba.ac.jp/iplab/paper/international/ali-idpt98.ps.gz

Using Concurrent Haskell to Develop Views over an.. - Einar W. Karlsen.. (1997) (Correct)
its support for attributed and versioned **objects**. The sixth section demonstrates the application
[Kar97b] extends Concurrent Haskell [PGF96] with a **message** passing model similar to the one of CML
The paper presents the higher order approach to **event** handling used within the WorkBench, as well as
www.informatik.uni-bremen.de/~ewk/papers/ifl97.ps.gz

A Short Course on Information Extraction: A Proposal - Bagga (1998) (Correct)
coreferences in text is trying to identify two **objects** (nouns) in the text which refer to the same
mechanism to analyze the performance of various **message** understanding systems [Bagga 1997d]Currently,
activities. Then the slot for the terrorist **event** type must be filled with one of these 24 possible
www.cs.ut.ee/~koit/SS98/proposal.ps

FUDGETS - Graphical User Interfaces and I/O in Lazy.. - Carlsson, Hallgren (1993) (Correct)
with menus, buttons and other graphical interface **objects**, without conforming to more or less imperative
in a hierarchical structure, and they interact by **message** passing. The current implementation is based on
:26 7 The internals of fudgets 27 7.1 Low level **events** and commands :
ftp.cs.chalmers.se/pub/cs-reports/papers/fudget-report/report.ps.gz

First 20 documents Next 20

Try your query at: Amazon Barnes & Noble Google (RI) Google (Web) CSB DBLP